

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-049389

(43)Date of publication of application : 20.02.1998

(51)Int.Cl.

G06F 9/46

G06F 12/00

(21)Application number : 08-223267

(71)Applicant : CASIO COMPUT CO LTD

(22)Date of filing : 07.08.1996

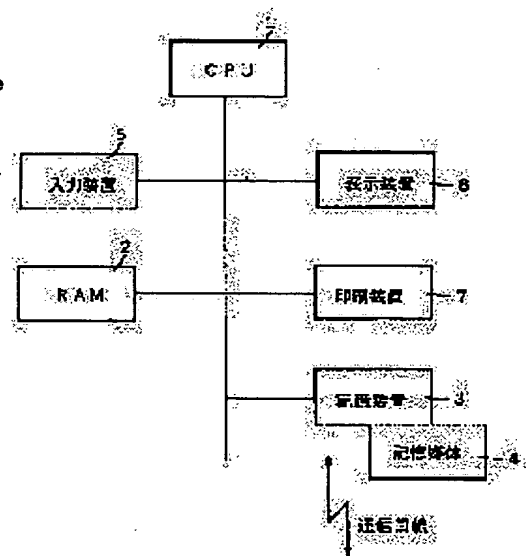
(72)Inventor : KURODA TETSUJI

## (54) PROGRAM MANAGEMENT DEVICE

### (57)Abstract:

**PROBLEM TO BE SOLVED:** To simplify the design of a processing program and also to deal with the change, addition, etc., of the program by deciding the presence or absence of an exclusive relation based on the read-out exclusive control information and the exclusive control information corresponding to the processing program under execution.

**SOLUTION:** When the start of a processing program is designated, a CPU 1 refers to an exclusive control definition file to read out the exclusive control information corresponding to the processing program. Then the CPU 1 decides the presence or absence of an exclusive relation based on the exclusive control information on the processing program whose start is designated this time and the exclusive control information on the processing program under execution. If no exclusive relation is confirmed, the execution is allowed to the designated processing program. If the exclusive relation is confirmed, an 'exclusive control operation' is read out of the exclusive control information corresponding to the processing program. Then the program execution is controlled according to the exclusive control operation. Thus, the exclusive control can be smoothly carried of for an entire system.



## LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平 1 0 - 4 9 3 8 9

(43) 公開日 平成 1 0 年 ( 1 9 9 8 ) 2 月 2 0 日

(51) Int. Cl. <sup>6</sup>	識別記号	庁内整理番号	F I	技術表示箇所
G06F 9/46	340		G06F 9/46	340 F
12/00	535		12/00	535 Z

審査請求 未請求 請求項の数 6 F D (全 1 0 頁)

(21) 出願番号 特願平 8 - 2 2 3 2 6 7

(22) 出願日 平成 8 年 ( 1 9 9 6 ) 8 月 7 日

(71) 出願人 0 0 0 0 0 1 4 4 3

カシオ計算機株式会社

東京都新宿区西新宿 2 丁目 6 番 1 号

(72) 発明者 黒田 哲司

東京都羽村市栄町 3 丁目 2 番 1 号 カシオ

計算機株式会社羽村技術センター内

(74) 代理人 弁理士 杉村 次郎

(54) 【発明の名称】 プログラム管理装置

(57) 【要約】 (修正有)

【課題】 プログラム設計時に個々の処理プログラムの中にシステム全体の排他関係を考慮に入れた情報を記述しておかなくても、個々の処理プログラムが共用資源をどのように利用するかを示す情報を各処理プログラムに対応付けて別個独立に定義しておくだけで、システム全体の排他制御を円滑に実行することができ、処理プログラムの設計を簡素化すると共に、プログラムの変更、追加等にも対応する。

【解決手段】 排他制御定義ファイルには処理プログラムとは別個独立に定義され、処理プログラムに 1 : 1 に対応付けて排他制御情報が記憶されている。処理プログラムの起動が指定されると、CPU はそれに対応する排他制御情報と現在実行中の排他制御情報とに基づいて排他関係の有無をチェックする。ここで、排他関係が有れば、指定プログラムの実行を制御する。

排他制御定義ファイル

プログラム ID	使用ファイル名	ファイル利用形態	排他事項	排他制御動作
P001	商品マスタ を 読 出 せ	参照 参照	更新 更新	実行中止 実行中止
P002	商品マスタ を 読 出 せ	参照 更新	更新 参照および更新	ウェイト ウェイト

1

## 【特許請求の範囲】

【請求項 1】 同時並行的に実行可能な複数の処理プログラムによって共用資源をアクセスする際に、排他制御を行って処理プログラムの実行を制御することにより、共用資源の競合を排除するプログラム管理装置であって、前記複数の処理プログラム毎に、処理プログラムとは別個独立に定義されており、共用資源を当該処理プログラムがどのように利用するかを示す情報を排他制御情報として記憶する制御情報記憶手段と、

前記複数の処理プログラムのうちそのいずれかの起動が指定された際に、前記制御情報記憶手段から当該指定処理プログラムに対応する排他制御情報を読み出す読み出し手段と、

この読み出し手段によって読み出された排他制御情報と現在実行中の処理プログラムに対応する排他制御情報とに基づいて排他関係の有無を判別する判別手段と、

この判別手段によって排他関係が有ることが判別された際に、前記指定処理プログラムの実行を制御するプログラム制御手段とを具備したことを特徴とするプログラム管理装置。

【請求項 2】 前記排他制御情報は、それに対応する処理プログラムを示すプログラム識別データおよび当該処理プログラムによってアクセスされる共用資源の名称を定義する他に、この共用資源を利用する処理名を利用形態として定義すると共に、この利用形態に相反する処理名を排他事項として定義する制御情報であり、

前記プログラム制御手段は、前記指定処理プログラムの排他制御情報と、現在実行中の処理プログラムに対応する排他制御情報とに基づいて排他関係の有無を判別する際に、同一の共用資源名に対応する前記利用形態と排他事項とを比較することによって排他関係の有無を判別するようにしたことを特徴とする請求項 1 記載のプログラム管理装置。

【請求項 3】 前記排他制御情報内に排他制御時の具体的な動作内容が排他制御動作として定義している場合に、前記プログラム制御手段は当該排他制御動作にしたがって前記指定処理プログラムの実行を制御するようにしたことを特徴とする請求項 1 記載のプログラム管理装置。

【請求項 4】 前記処理プログラム毎に、その排他制御情報が入力された際に、入力された排他制御情報を処理プログラムに対応付けて前記制御情報記憶手段に設定する設定手段を設けたことを特徴とする請求項 1 記載のプログラム管理装置。

【請求項 5】 前記処理プログラムの記述内容を解析することによって排他制御情報を自動生成する生成手段と、この生成手段によって生成された排他制御情報を当該処理プログラムに対応付けて前記制御情報記憶手段に設定する設定手段とを設けたことを特徴とする請求項 1 記載のプログラム管理装置。

【請求項 6】 前記生成手段は、前記処理プログラムの中

2

に共用資源に対するアクセスが記述されている場合に、そのアクセス対象を共用資源名として求め、かつ、アクセスの種類が共用資源に対する読み出しか、書き込みかに応じて共用資源を利用する処理名を利用形態として求めると共に、この利用形態に相反する処理名を排他事項として求め、前記共用資源名、利用形態、排他事項を含む排他制御情報を自動生成するようにしたことを特徴とする請求項 5 記載のプログラム管理装置。

## 【発明の詳細な説明】

## 【 0 0 0 1 】

【発明の属する技術分野】 この発明は、オフィスコンピュータ、パーソナルコンピュータや通信システム等において、同時並行的に複数の処理プログラムによって共用資源をアクセスする際に、排他制御によってプログラムの実行を制御するプログラム管理装置に関する。

## 【 0 0 0 2 】

【従来の技術】 一般に、マルチタスク処理を行うオフィスコンピュータ、パーソナルコンピュータ等のデータ処理装置においては、同時並行的に実行可能な複数の処理プログラムによって共用資源（他のプログラムやデータベース）をアクセスする際に、ファイルロックやレコードロックと呼ばれる排他制御によってプログラムの実行を制御し、共用資源の競合を排除するようにしている。この場合、ファイルロックやレコードロックについての制御情報は、予め個々の処理プログラム内に記述するようにしていた。

## 【 0 0 0 3 】

【発明が解決しようとする課題】 このため、システムエンジニア等のプログラム作成者は、プログラム設計時にシステムの全体構成を完全に理解し、どのような排他関係を処理プログラム内に記述する必要があるかを把握した上でプログラムを設計しなければならず、プログラム設計が極めて困難なものとなる要因となっていた。また、システムの拡張や仕様変更等によって新しいプログラムが追加されたり、既存のプログラムが変更された場合には、再度、システム全体の排他関係を見直し、変更による影響関係を見極めた上で対処する必要があった。したがって、高度なシステム設計能力を要求されたり、設計ミスによってデッドロックが発生するというおそれがあった。この発明の課題は、プログラム設計時に個々の処理プログラムの中にシステム全体の排他関係を考慮に入れた情報を記述しておかなくても、個々の処理プログラムが共用資源をどのように利用するかを示す情報を各処理プログラムに対応付けて別個独立に定義しておくだけで、システム全体の排他制御を円滑に実行することができ、処理プログラムの設計を簡素化すると共に、プログラムの変更、追加等にも対応できるようにすることである。

## 【 0 0 0 4 】

【課題を解決するための手段】 この発明の手段は次の通

10

20

30

40

50

りである。同時並行的に実行可能な複数の処理プログラムによって共用資源をアクセスする際に、排他制御を行って処理プログラムの実行を制御することにより、共用資源の競合を排除するプログラム管理装置であって、

( 1 )、制御情報記憶手段は前記複数の処理プログラム毎に、処理プログラムとは別個独立に定義されており、共用資源を当該処理プログラムがどのように利用するかを示す情報を排他制御情報として記憶する。

( 2 )、読み出し手段は前記複数の処理プログラムのうちそのいずれかの起動が指定された際に、前記制御情報記憶手段から当該指定処理プログラムに対応する排他制御情報を読み出す。

( 3 )、判別手段はこの読み出し手段によって読み出された排他制御情報と現在実行中の処理プログラムに対応する排他制御情報とに基づいて排他関係の有無を判別する。

( 4 )、プログラム制御手段はこの判別手段によって排他関係が有ることが判別された際に、前記指定処理プログラムの実行を制御する。なお、前記排他制御情報は、それに対応する処理プログラムを示すプログラム識別データおよび当該処理プログラムによってアクセスされる共用資源の名称を定義する他に、この共用資源を利用する処理名を利用形態として定義すると共に、この利用形態に相反する処理名を排他事項として定義する制御情報であり、前記プログラム制御手段は、前記指定処理プログラムの排他制御情報と、現在実行中の処理プログラムに対応する排他制御情報とに基づいて排他関係の有無を判別する際に、同一の共用資源名に対応する前記利用形態と排他事項とを比較することによって排他関係の有無を判別するようにしてもよい。また、前記排他制御情報内に排他制御時の具体的な動作内容が排他制御動作として定義している場合に、前記プログラム制御手段は当該排他制御動作にしたがって前記指定処理プログラムの実行を制御するようにしてもよい。また、前記処理プログラム毎に、その排他制御情報が入力された際に、入力された排他制御情報を処理プログラムに対応付けて前記制御情報記憶手段に設定する設定手段を設けてもよい。更に、前記処理プログラムの記述内容を解析することによって排他制御情報を自動生成する生成手段と、この生成手段によって生成された排他制御情報を当該処理プログラムに対応付けて前記制御情報記憶手段に設定する設定手段とを設けてもよい。この場合、前記生成手段は、前記処理プログラムの中に共用資源に対するアクセスが記述されている場合に、そのアクセス対象を共用資源名として求め、かつ、アクセスの種類が共用資源に対する読み出しか、書き込みかに応じて共用資源を利用する処理名を利用形態として求めると共に、この利用形態に相反する処理名を排他事項として求め、前記共用資源名、利用形態、排他事項を含む排他制御情報を自動生成するようにしてもよい。

【 0 0 0 5 】いま、同時並行的に実行可能な複数の処理プログラム毎に、処理プログラムとは別個独立に排他制御情報が定義されているものとする。この排他制御情報は、それに対応する処理プログラムが共用資源をどのように利用するかを示す情報であり、システム全体の排他関係を考慮に入れることなく、例えば、システムエンジニア等が個々の処理プログラムに対応付けて入力設定されたもの、あるいは処理プログラムを自動解析することによって生成されたものである。ここで、処理プログラムの起動が指定されると、それに対応する排他制御情報と現在実行中の処理プログラムに対応する排他制御情報とに基づいて排他関係の有無が判別され、排他関係があれば、指定処理プログラムの実行が制御される。したがって、プログラム設計時に個々の処理プログラムの中にシステム全体の排他関係を考慮に入れた情報を記述しておかなくても、個々の処理プログラムが共用資源をどのように利用するかを示す情報を各処理プログラムに対応付けて別個独立に定義しておくだけで、システム全体の排他制御を円滑に実行することができ、処理プログラムの設計を簡素化すると共に、プログラムの変更、追加等にも対応することができる。

【 0 0 0 6 】

【発明の実施の形態】

( 第 1 実施形態 ) 以下、図 1 ～図 6 を参照してこの発明の第 1 実施形態を説明する。図 1 はオフィスコンピュータやパーソナルコンピュータ等のデータ処理装置を示したブロック構成図である。CPU 1 は RAM 2 内の各種プログラムにしたがってこのデータ処理装置の全体動作を制御する中央演算処理である。RAM 2 に対しては記憶装置 3 からプログラム、データ等を読み込んでおく。記憶装置 3 はプログラム、データ等が予め記憶されている記憶媒体 4 を有しており、この記憶媒体 4 は磁気的、光学的記憶媒体、もしくは半導体メモリで構成されている。この記憶媒体 4 は記憶装置 3 に固定的に設けたもの、もしくは着脱自在に装着するものである。また、前記記憶媒体 4 に記憶するプログラム、データ等は、通信回線等を介して接続された他の機器から受信して記憶する構成にしてもよく、更に、通信回線等を介して接続された他の機器側に前記記憶媒体 4 を備えた記憶装置を設け、この記憶媒体 4 に記憶されているプログラム、データを通信回線を介して使用する構成にしてもよい。

【 0 0 0 7 】入力装置 5 は各種のデータやコマンドを入力するキー入力装置とマウス等のポインティングデバイスを有する構成で、CPU 1 は入力装置 5 から入力されたデータを取り込んで表示装置 6 に表示出力させたり、RAM 2 に書き込む。また、入力装置 5 から印刷指令が入力されると、CPU 1 は RAM 2 から印刷対象のデータを読み出し、印刷装置 7 を起動させて印字出力させる。

【 0 0 0 8 】図 2 は RAM 2 内に展開されるファイルの

一部を示した構成図で、RAM 2 はプログラムファイル 2-1、データファイル 2-2、排他制御定義ファイル 2-3 を有する構成となっている。プログラムファイル 2-1 は記憶装置 3 からロードされた各種のプログラムを記憶するもので、CPU 1 は同時並行的に実行可能な複数の処理プログラムによって共用資源（他の処理プログラムやデータファイル）をアクセスする際に、排他制御によって共用資源の競合を排除するようにしている。なお、各処理プログラムには排他制御についての情報は記述されておらず、それを処理プログラムとは別個独立させ、後述する排他制御定義ファイル 2-3 に処理プログラムに 1 : 1 に対応付けて排他制御情報を設定するようにしている。データファイル 2-2 は記憶装置 3 からロードされた商品マスタファイル、在庫台帳ファイル等のデータのデータファイルを記憶するもので、処理プログラムの実行中に CPU 1 はデータファイル 2-2 をアクセスしてファイル処理を実行する。

【0009】図 3 は排他制御定義ファイル 2-3 の構成を示したもので、排他制御定義ファイル 2-3 は複数の処理プログラム毎に、処理プログラムが共用資源をどのように利用するかを示す情報を排他制御情報として記憶するデータファイルである。排他制御情報は、処理プログラムに 1 : 1 に対応付けられたもので、その処理プログラム名を示すプログラム ID および当該処理プログラムによってアクセスされる共用資源（データファイル）の名称（使用ファイル名）の他に、この共用資源を利用する処理名を示すファイル利用形態と、このファイル利用形態に相反する処理名を示す排他事項と、排他制御の具体的な動作内容（排他制御動作）を定義する。ここで、プログラム ID（P001）で示される処理プログラムは、“在庫数照会”のような参照型のプログラムであり、使用ファイル名として“商品マスタ”、“在庫台帳”がそれぞれ定義されている。この場合、各使用ファイル名に対応するファイル利用形態としてそれぞれ“参照”が定義され、また、排他事項としてそれぞれ“更新”が定義され、更に、排他制御動作としてそれぞれ“実行中止”が定義されている。また、プログラム ID（P002）で示される処理プログラムは、“入出庫処理”のように在庫情報を更新する更新型のプログラムであり、使用ファイル名として“商品マスタ”、“在庫台帳”がそれぞれ定義されている。この場合、ファイル利用形態として商品マスタについては“参照”、在庫台帳については“更新”が定義され、また、排他事項として商品マスタについては“更新”、在庫台帳については“参照および更新”が定義され、更に、排他制御動作としては商品マスタ、在庫台帳にそれぞれ対応して“ウェイト”が定義されている。

【0010】このように構成された排他制御定義ファイル 2-3 を参照し、CPU 1 は処理プログラムの起動が指定されると、その処理プログラムに対応する排他制御

情報を読み出す。ここで、CPU 1 は今回起動が指定された指定処理プログラムの排他制御情報と、現在実行中の処理プログラムの排他制御情報とに基づいて排他関係の有無を判別し、排他関係が無ければ、当該指定処理プログラムの実行を許可するが、排他関係があれば、そのプログラムに対応する排他制御情報から“排他制御動作”を読み出してそれに応じてプログラムの実行を制御する。

【0011】次に、このデータ処理装置における排他制御動作を図 4 に示すフローチャートにしたがって説明する。なお、図 4 およびその他のフローチャートに記述されている各機能を実現するためのプログラムは、CPU 1 が読み取り可能なプログラムコードの形態で記憶媒体 4 に記憶されており、その内容が RAM 2 内のプログラムファイル 2-1 にロードされている。入力装置 5 から処理プログラムの起動が指示されると、図 4 に示す排他制御処理が実行開始される。いま、排他制御定義ファイル 2-3 には予め各種の処理プログラムに対応付けてその排他制御情報が入力設定されているものとする。この場合、入力装置 5 から排他制御情報を入力する他に、記憶媒体 4 あるいは通信回線を介して外部から供給するようにしてもよい。

【0012】まず、CPU 1 は起動指定された処理プログラムの ID を取り込み、このプログラム ID に基づいて排他制御定義ファイル 2-3 を検索し（ステップ A1）、それに対応する排他制御情報を読み出す（ステップ A2）。そして、現在実行中の他の処理プログラムが存在するかをチェックする（ステップ A3）。この場合、実行中のプログラムが無ければ、共用資源の競合は当然、発生しないので、起動指定された処理プログラムを無条件に実行開始させる（ステップ A6）。一方、他に処理プログラムが実行中であれば（ステップ A3）、排他関係のチェックを行う（ステップ A4）。すなわち、今回起動指定された処理プログラムと現在実行中の処理プログラムとは共用資源を競合するプログラム同士か否かに拘らず、現在実行中の処理プログラムがあれば、排他関係の有無をチェックするようにしている。

【0013】図 5 はこの排他関係チェック処理を示したフローチャートである。まず、起動指定された処理プログラムに対応する排他制御情報を RAM 2 内のワーク域 A にセットする（ステップ B1）。そして、このワーク域 A をアドレス指定するポインタ i をクリアしたのち（ステップ B2）、その値に「1」を加算するインクリメント処理を行う（ステップ B3）。そして、このポインタ i でアドレス指定されるワーク域 A から使用ファイル名を読み出し、このファイル名が現在実行中の他の処理プログラムで利用されているかをチェックする（ステップ B4）。いま、プログラム ID（P001）で示される処理プログラムの起動が指定された際には、そのファイル名として“商品マスタ”が読み出され、この商品

マスタファイルが現在実行中の他の処理プログラムで利用されているかを調べる。ここで、商品マスタファイルが利用されていないければ、そのファイル名がワーク域A内の最終ファイル名でないことを条件に（ステップB6）、ポインタiを更新する（ステップB3）。

【0014】これによって、次のファイル名として“在庫台帳”が読み出され、このファイル名が現在実行中の他の処理プログラムで利用されている場合には（ステップB5）、その処理プログラムに対応する排他制御情報から“利用形態”を読み出し、これをRAM2内のワーク域Bにセットする（ステップB7）。そして、ワーク域B内の“利用形態”とポインタiで指定されるワーク域A内の“排他事項”とを比較し、利用形態が排他事項内に含まれているかを調べる（ステップB8）。ここで、両者の不一致が検出されると、ステップB6に進むが、両者の一致が検出されると、排他関係が有ると判断し、ポインタiで示されるワーク域Aから“排他制御動作”を読み出す（ステップB9）。なお、ステップB6で最終ファイル名であることが検出された場合には、起動指定された処理プログラムによって利用されるファイルは、現在どの処理プログラムでも利用されていない場合であり、排他関係が無いことが検出される。

【0015】このようにして排他関係のチェックが終ると、図4のステップA5に進み、そのチェック結果（排他関係の有無）を判別し、起動指定された処理プログラムの実行が可能かを調べる。ここで、排他関係が無ければ、当該指定プログラムの実行を開始するが（ステップA6）、排他関係が有れば、読み出された“排他制御動作”にしたがって指定処理プログラムの実行を制御する（ステップA7）。この場合、指定処理プログラムの実行を中止したり、排他関係が無くなるまでそのプログラムの実行を待機させる。

【0016】以上のように構成されたデータ処理装置においては、個々の処理プログラムに1:1に対応させて“プログラムID”、“使用ファイル名”、“ファイル利用形態”、“排他事項”、“排他制御動作”から成る排他制御情報を別個独立して定義しておくだけで、システム全体についての排他制御を円滑に実行することが可能となる。したがって、プログラム設計時に個々の処理プログラムの中にシステム全体の排他関係を考慮に入れた情報を記述する必要がなくなるため、処理プログラムの設計が極めて簡単なものとなり、処理プログラムの変更、追加等にも対応することができるようになる。

【0017】（第2実施形態）以下、図6を参照してこの発明の第2実施形態を説明する。なお、第1実施形態においては、処理プログラムの起動時に共用資源の競合を排除する排他制御をファイル単位で行うようにしたが、第2実施形態は共用資源の競合をレコード単位やフィールド単位で制御できるようにしたものである。図6は第2実施形態における排他制御定義ファイル2-3の

内容を示したもので、そのファイル利用形態および排他事項に、レコード単位、フィールド単位で参照、更新といった情報を設定することにより、ファイルロックだけではなく、レコードロックやフィールドロックというきめ細かな排他制御も実現することができる。

【0018】すなわち、第1実施形態においてプログラムID（P001）、（P002）で示される各処理プログラムは、ファイル単位で同一共用資源を利用するため、それらを同時に実行することはできないが、第2実施形態では同一ファイルであっても利用するレコード、フィールドが異なれば、同時実行が可能となる。例えば、プログラムID（P001）で示される処理プログラムの起動時において、その処理プログラムは商品マスタファイル内の“商品名”というフィールドのみを参照するものであり、他の処理プログラムがそのフィールド（商品名）を更新中であれば、処理プログラム（P001）はそのフィールド内容の更新が終わるまでウェイト状態となる。一方、他の処理プログラムが当該フィールド（商品名）を除く他のフィールドを更新中であれば、処理プログラム（P001）はそのフィールド内容を参照することができる。

【0019】（第3実施形態）以下、図7に示すフローチャートに基づいてこの発明の第3実施形態を説明する。なお、第1実施形態は排他制御定義ファイル2-3の内容を入力装置5あるいは記憶媒体4等から入力設定するようにしたが、この第3実施形態は処理プログラムの記述内容を解析することによって排他制御情報を自動生成し、これを排他制御定義ファイル2-3に設定するようにしたものである。図7は排他制御情報を自動生成して排他制御定義ファイル2-3に設定する場合の動作を示したフローチャートであり、排他制御定義ファイル2-3に対する設定処理を入力指定すると、図7のフローチャートが実行開始される。まず、処理プログラムに対応して排他制御情報を設定すべき処理プログラムを入力装置5から選択的に指定すると（ステップC1）、CPU1は当該処理プログラムを記憶装置3からプログラムファイル2-1にロードすると共に、そのプログラムIDを排他制御情報として求める（ステップC2）。このように指定処理プログラムがロードされている状態において、この処理プログラムの種類を入力指定する（ステップC3）。この場合、処理プログラムの種類として、入力データに基づいてデータ処理を行う対話型のプログラムか否かの指定を行う。ここで、対話型の処理プログラムが指定された場合（ステップC4）、CPU1は排他制御情報内に定義される「排他制御動作」として“実行中止”を生成するが（ステップC5）、対話型の処理プログラムではなければ、「排他制御動作」として“ウェイト”を生成する（ステップC6）。

【0020】次に、ステップC7に進み、指定処理プログラムを構成するソースコードの先頭ステップをアクセ

スしてそれを読み出し、その記述がデータファイルのアクセスで無ければ(ステップC 8)、次のソースコードを読み込むためにステップC 1 5に進み、ソースコードを全て読み終えたかをチェックする(ステップC 1 6)。いま、終了でなければ、ステップC 8に戻る。ここで、ファイルアクセスが記述されていることが検出されると、アクセス対象のファイル名を排他制御情報に定義される「使用ファイル名」として読み込む(ステップC 9)。続いて、アクセスの種類がデータファイルに対する読み出しか、書き込みかを求め、排他制御情報に定義される「ファイル利用形態」としてリードの場合には“参照”ライトの場合には“更新”を生成する(ステップC 1 0)。更にアクセスの種類を判別する(ステップC 1 1)。いま、データファイルに対するリードが記述されている場合には、排他制御情報に定義される「排他事項」として“更新”を生成するが(ステップC 1 2)、データファイルに対するライトが記述されていれば、「排他事項」として“参照および更新”を生成する(ステップC 1 3)。このようにして生成された「プログラムID」、「使用ファイル名」、「ファイル利用形態」、「排他事項」、「排他制御動作」をそれぞれ排他制御情報として排他制御定義ファイル2-3に設定する(ステップC 1 4)。以上のように処理プログラムの記述内容を解析することによって「プログラムID」、「使用ファイル名」、「ファイル利用形態」については直接、プログラム記述から自動生成することができる。この場合、「排他事項」、「排他制御動作」については一定の規則に基づいて生成されたもので、例えば、利用形態が“参照”ならば排他事項を“更新”とする。また、“排他制御動作”については、処理プログラムの種類が入力指定された際に、その種類が対話型であれば、“実行中止”とする。なお、「ファイル利用形態」、「排他事項」についてはファイル単位に限らず、レコード単位やフィールド単位でも自動生成することができる。このように排他制御情報はプログラム記述から自動生成されて排他制御定義ファイル2-3に設定されるので、その設定を効率良くしかも確実に行うことが可能となる。

【0021】なお、スタンド・アロンタイプのデータ処理装置に限らず、サーバ/クライアントシステム等の通信システムにも適用可能である。この場合、サーバ側コンピュータがクライアント側コンピュータに排他制御定義ファイル2-3を処理プログラムと共に提供するようにしてもよい。

【0022】

【発明の効果】この発明によれば、プログラム設計時に個々の処理プログラムの中にシステム全体の排他関係を考慮に入れた情報を記述しておかなくても、個々の処理プログラムが共用資源をどのように利用するかを示す情報を各処理プログラムに対応付けて別個独立に定義しておくだけで、システム全体の排他制御を円滑に実行することができ、処理プログラムの設計を簡素化すると共に、プログラムの変更、追加等にも対応することができる。これによって、処理プログラムの設計を大幅に簡素化することができると共に、プログラムの変更や追加等にも対応することが可能となる。また、排他制御情報の設定には、高度なシステム設計能力が要求されないもので、システムエンジニア等の専門家にその設定を依頼しなくても、業務処理において共用資源をどのように利用するかを熟知している一般の業務担当者であれば、その設定を容易に行うことができ、コスト的にもメンテナンス的にも極めて有利なものとなる。

【図面の簡単な説明】

【図1】オフィスコンピュータ、パーソナルコンピュータ等のデータ処理装置の全体構成を示したブロック図。

【図2】RAM 2の内容を示した図。

【図3】排他制御定義ファイル2-3の内容を示した図。

【図4】排他制御処理を示したフローチャート。

【図5】図4のステップA 4(排他関係チェック処理)を示したフローチャート。

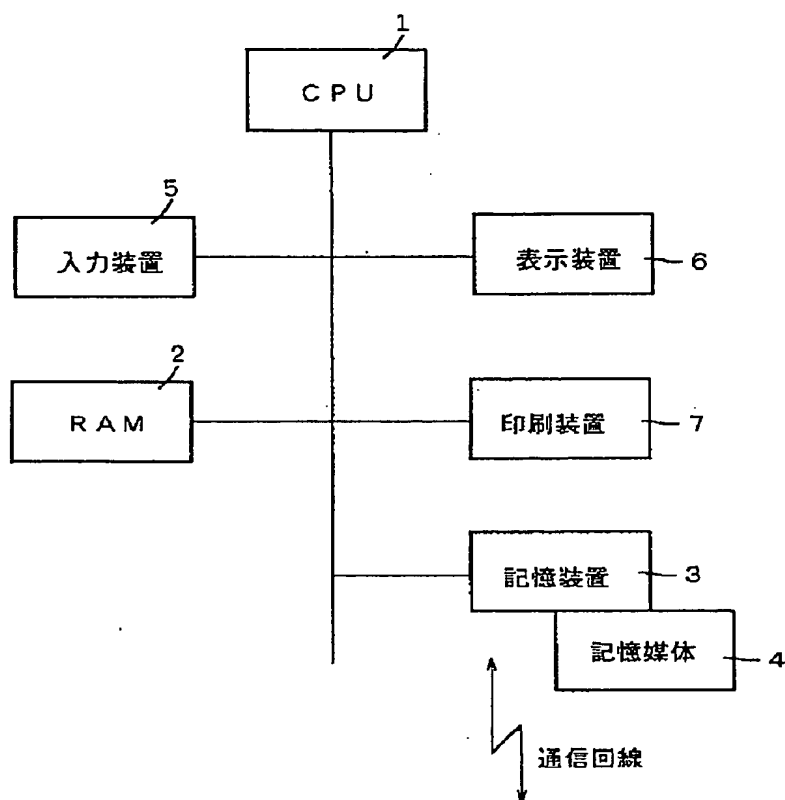
【図6】第2実施形態を説明するための排他制御定義ファイル2-3の内容を示した図。

【図7】第3実施形態において、処理プログラムの記述内容を解析することによって排他制御情報を自動生成して排他制御定義ファイル2-3に設定する場合の動作を示したフローチャート。

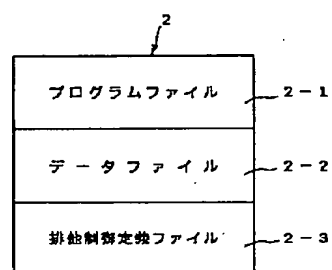
【符号の説明】

- 1 CPU
- 2 RAM
- 2-1 プログラムファイル
- 2-2 データファイル
- 2-3 排他制御定義ファイル
- 3 記憶装置
- 4 記憶媒体
- 5 入力装置
- 6 表示装置

【図 1】



【図 2】



【図 3】

排他制御定義ファイル

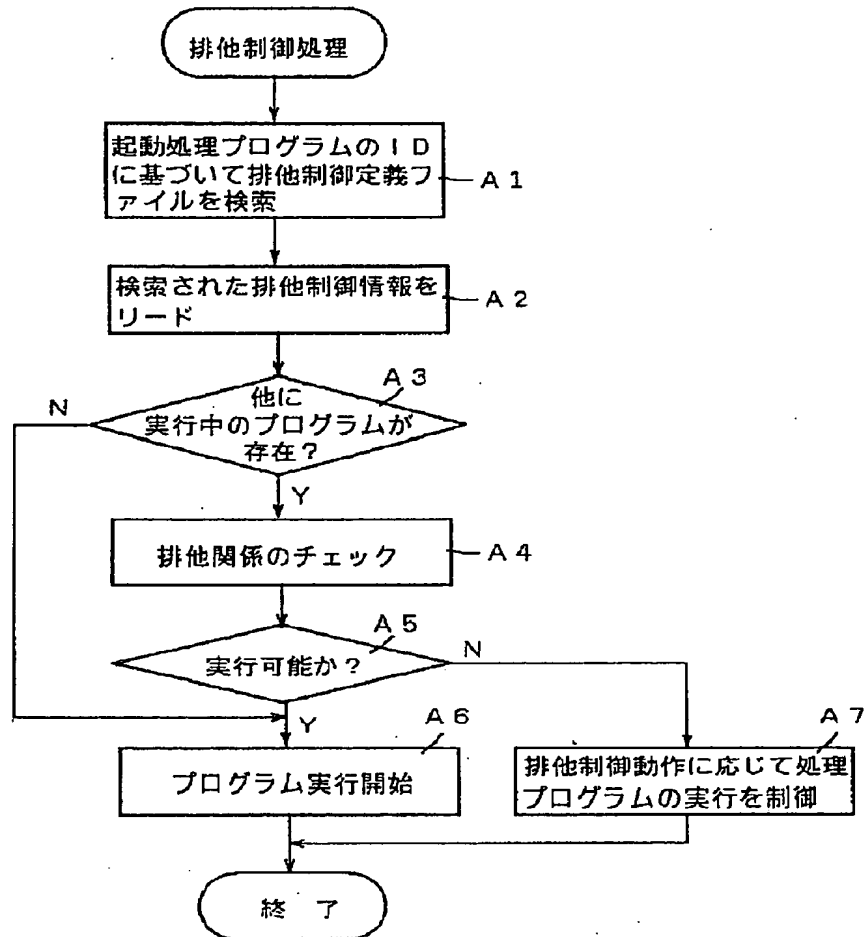
プログラムID	使用ファイル名	ファイル利用形態	排他事項	排他制御動作
P001	商品マスタ 在庫台帳	参照 参照	更新 更新	実行中止 実行中止
P002	商品マスタ 在庫台帳	参照 更新	更新 参照および更新	ウェイト ウェイト

【図 6】

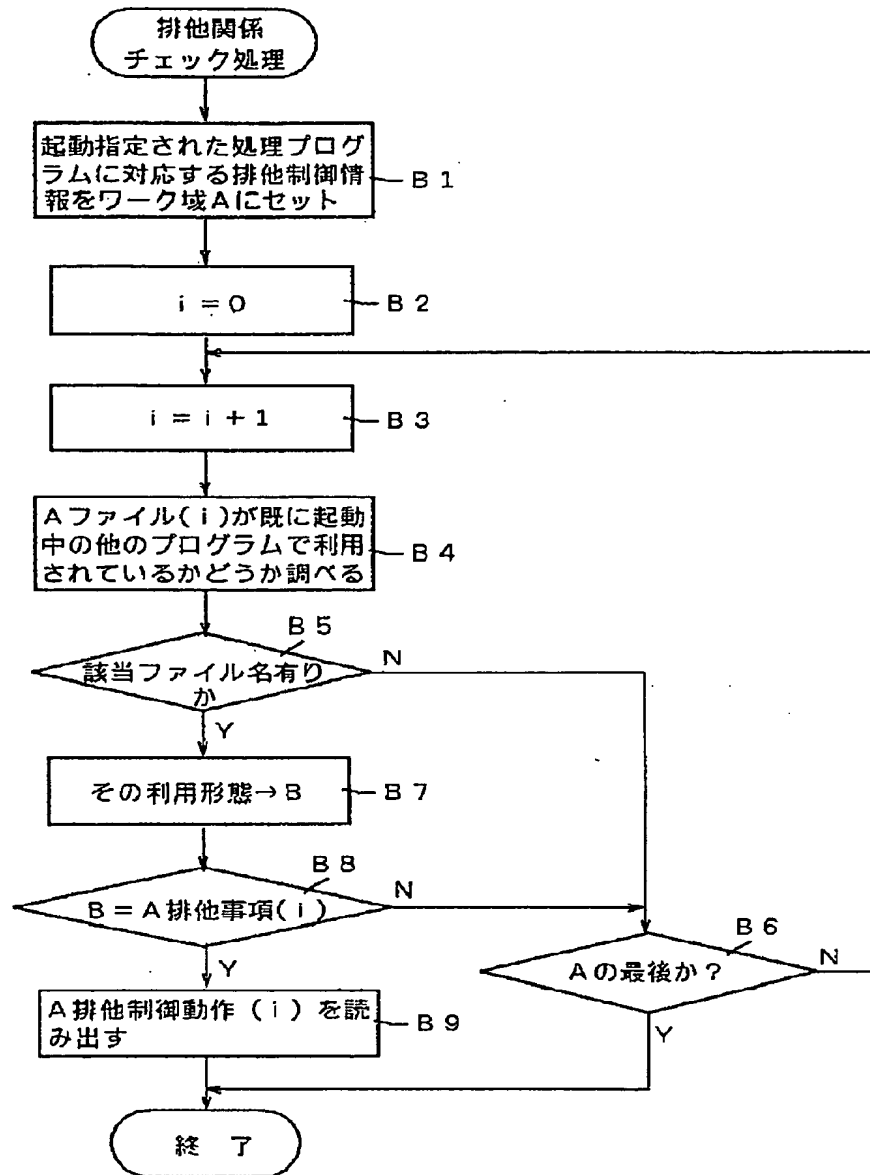
プログラムID	使用ファイル名	ファイル利用形態	排他事項	排他制御動作
P001	商品マスタ 在庫台帳	フィールド参照 (商品名) レコード参照	参照フィールドの更新 参照レコードの更新	解除されるまでウェイト 解除されるまでウェイト
P002	商品マスタ 在庫台帳	フィールド参照 (商品名) レコード更新	参照フィールドの更新 参照レコードの参照および更新	解除されるまでウェイト プログラムの実行を中止する



【図 4】



【図 5】



【 図 7 】

